APPENDIX A:

BUFFER MANAGER INTERFACE

CBUS Interface and Registers

In certain embodiments of the invention, CBUS interface is shared with the I/O core interface. Table A1 provides a description of I/O signals between the CBUS interface and I/O core 290.

TABLE A1: CBUS interface signals between I/O core module

Number	Signal	I/O	Description
1	CbusAD_BM[31:0]	Input	CBUS Address/Data bus to register
2	Cbus_valid	Input	CBUS command valid
3	CbusWr_NotRd	Input	CBUS write / read 1 : write mode 0 : read mode
4	BM_Rdata[31:0]	Output	Register data return to CBUS.

Table A2 provides a summary of registers included in CBUS interface and registers 260.

TABLE A2: Register summary

Address offset	Register	Description
0	Threshold register 1	
1	Threshold register 2	
2	Threshold register 3	
3	Data buffer base address	
4	Interrupt status	
5	Interrupt enable	
6	Control	
7	RFQ	
8	RRQ	
9	TRQ 1	
10	TRQ 2	
11	TRQ 3	
12	TRQ 4	
13	WFQ	
14	DMA command	
15	BRQ expansion counters	

Address offset	Register	Description
16	TRQ 1 expansion counters	
17	TRQ 2 expansion counters	
18	TRQ 3 expansion counters	
19	TRQ 4 expansion counters	
20	Current DMA	
21	FIFO status FIFO pointer	
22	FIFO status data	
23	TRQ pointer read	
24	BRQ frame counter	

Tables A3 through A26 provide a detailed description of registers included in CBUS interface and registers 260, in accordance with one embodiment of the invention.

TABLE A3: Threshold register 1

Bit field	Name	R/W	default	Description
15:8	BRQ FIFO threshold	R/W	0	
7:0	Read Free FIFO threshold	R/W	0	

In embodiments of the system, TRQ write FIFO and TRQ read FIFO use same threshold value set by this register.

TABLE A4: Threshold register 2

Bit field	Name	R/W	default	Description
31:24	TRQ 4 FIFO threshold	R/W	0	
23:16	TRQ 3 FIFO threshold	R/W	0	
15:8	TRQ 2 FIFO threshold	R/W	0	
7:0	TRQ 1 FIFO threshold	R/W	0	

TABLE A5: Threshold register 3

Bit field	Name	R/W	default	Description
31:24	Write Free FIFO 4 threshold	R/W	0	
23:16	Write Free FIFO 3 threshold	R/W	0	
15:8	Write Free FIFO 2 threshold	R/W	0	

Bit field	Name	R/W	default	Description
7:0	Write Free FIFO 1 threshold	R/W	0	

TABLE A6: Data buffer base address register

Bit field	Name	R/W	default	Description
8:0	Data buffer base address	R/W	0	Upper 9 bits of base address

The buffer manager conveys information to the CPU and requests actions from the CPU by means of one or more interrupt signals requests (ISR). The interrupt status is latched until cleared by the CPU. Writing "1" to one or more of the bits in the interrupt status register will reset (set to "0") that status bit. If the interrupt condition still remains, that bit will be set again. Writing "0" does not do anything. RRQ interrupt is set to "1" when RRQ has data and "frame_cmp" input is set to "1". FIFO illegal condition interrupt is set to "1" when FIFO overflow / underflow occurs, or MSB of expansion counter goes "1". This interrupt is reset when FIFO illegal condition interrupt bit (bit[17]) is set to "1". All the other interrupts are set to "1" when FIFO level matches threshold level.

TABLE A7: Interrupt status register

Bit field	Name	R/W	default	Description
17	FIFO illegal condition interrupt	R	0	1: interrupt 0: no interrupt
16	Frame counter interrupt	R	0	1: interrupt 0: no interrupt
15	BRQ Ready FIFO data ready interrupt	R	0	1: interrupt 0: no interrupt
14	Write Free FIFO 4 interrupt	R	0	1: interrupt 0: no interrupt
13	Write Free FIFO 3 interrupt	R	0	1: interrupt 0: no interrupt
12	Write Free FIFO 2 interrupt	R	0	1: interrupt 0: no interrupt
11	Write Free FIFO 1 interrupt	R	0	1: interrupt 0: no interrupt
10	TRQ 4 read FIFO interrupt	R	0	1: interrupt 0: no interrupt
9	TRQ 4 write FIFO interrupt	R	0	1: interrupt 0: no interrupt
8	TRQ 3 read FIFO interrupt	R	0	1: interrupt 0: no interrupt
7	TRQ 3 write FIFO interrupt	R	0	1: interrupt 0: no interrupt

Bit field	Name	R/W	default	Description
6	TRQ 2 read FIFO interrupt	R	0	1: interrupt 0: no interrupt
5	TRQ 2 write FIFO interrupt	R	0	1: interrupt 0: no interrupt
4	TRQ 1 read FIFO interrupt	R	0	1: interrupt 0: no interrupt
3	TRQ 1 write FIFO interrupt	R	0	1: interrupt 0: no interrupt
2	RFQ FIFO interrupt	R	0	1: interrupt 0: no interrupt
1	BRQ read FIFO interrupt	R	0	1: interrupt 0: no interrupt
0	BRQ write FIFO interrupt	R	0	1: interrupt 0: no interrupt

TABLE A8: Interrupt enable register

Bit field	Name	R/W	default	Description
17	FIFO illegal condition	R/W	0	1: interrupt enable 0: interrupt disable
16	Frame counter interrupt	R/W	0	1: interrupt enable 0: interrupt disable
15	BRQ FIFO data ready interrupt	R/W	0	1: interrupt enable 0: interrupt disable
14	WRQ FIFO 4 interrupt	R/W	0	1: interrupt enable 0: interrupt disable
13	WRQ FIFO 3 interrupt	R/W	0	1: interrupt enable 0: interrupt disable
12	WRQ FIFO 2 interrupt	R/W	0	1: interrupt enable 0: interrupt disable
11	WRQ FIFO 1 interrupt	R/W	0	1: interrupt enable 0: interrupt disable
10	TRQ 4 read FIFO interrupt	R/W	0	1: interrupt enable 0: interrupt disable
9	TRQ 4 write FIFO interrupt	R/W	0	1: interrupt enable 0: interrupt disable
8	TRQ 3 read FIFO interrupt	R/W	0	1: interrupt enable 0: interrupt disable
7	TRQ 3 write FIFO interrupt	R/W	0	1: interrupt enable 0: interrupt disable
6	TRQ 2 read FIFO interrupt	R/W	0	1: interrupt enable 0: interrupt disable
5	TRQ 2 write FIFO interrupt	R/W	0	1: interrupt enable 0: interrupt disable
4	TRQ 1 read FIFO interrupt	R/W	0	1: interrupt enable 0: interrupt disable
3	TRQ 1 write FIFO interrupt	R/W	0	1: interrupt enable 0: interrupt disable
2	RFQ FIFO interrupt	R/W	0	1: interrupt enable 0: interrupt disable

Bit field	Name	R/W	default	Description
1	BRQ read FIFO interrupt	R/W	0	1: interrupt enable 0: interrupt disable
0	BRQ write FIFO interrupt	R/W	0	1: interrupt enable 0: interrupt disable

According to one or more embodiments of the system, on power-up, the *lstate* is in "halt" state. Control software programs these 2 bits to "run" to start buffer management. Software can program these bits to 2'b10 to reset buffer management module. On completion of the reset sequence, the hardware will set *lstate* to "halt" (default). When a reset is issued all pending DMA commands in the queues will be flushed. DMA command that has already started (transfer in progress or memory request already issued by the DMA command sequencer) will be allowed to complete; the reset sequence will wait until the transfer is finished.

TABLE A9: Control register

Bit field	Name	R/W	default	Description
31:30	Lstate	R/W	2'b11	00:run; 11;halt; (default) 10:reset, reset entire buffer management and Interface
29:9	Reserved			
8:7	WFQ destination	R/W	0	When CPU write data pointer through WFQ register, these bits determine which queue to write. 00 : WFQ 1 01 : WFQ 2 10 : WFQ 3 11 : WFQ 4
6:5	TRQ configuration	R/W	0	00 : queue 1 ~ 4 = 32 entry 01 : queue 1 ~ 2 = 40 entry queue 3 ~ 4 = 24 entry 10 : queue 1 = 80 entry queue 2 ~ 3 = 24 entry 11 : queue 1 ~ 2 = 64 entry
4	TRQ 4 expansion	R/W	0	1 :Short circuit 0 :Automatic expansion
3	TRQ 3 expansion	R/W	0	1 :Short circuit 0 :Automatic expansion

Bit field	Name	R/W	default	Description
2	TRQ 2 expansion	R/W	0	1 :Short circuit 0 :Automatic expansion
1	TRQ 1 expansion	R/W	0	1 :Short circuit 0 :Automatic expansion
0	BRQ expansion	R/W	0	1 :Short circuit 0 :Automatic expansion

TABLE A10: RFQ register

Bit field	Name	R/ W	defaul t	Description
31:0	RFQ data	R	0	Data buffer pointer (read) Read all 1's if Read Free FIFO is empty

TABLE A11: RRQ register

Bit field	Name	R/W	default	Description
31:0	RRQ data	R	0	1st read : Lower word of BD 2nd read : Upper word of BD Read all 1's if Rx BD Ready read FIFO is empty

TABLE A12: TRQ register

Bit field	Name	R/W	default	Description
31:0	TRQ data	R/W	0	1st write : Lower word of BD 2nd write : Upper word of BD Read all 1's if Rx BD Ready write FIFO is full

Bit [22:7] of this 32 bit data will be written into WFQ FIFO selected by WFQ destination bit in control register.

TABLE A13: WFQ register

Bit field	Name	R/W	default	Description
31:0	WFQ data	R/W	0	Data buffer pointer (write) Read all 1's if Read Free FIFO is full

In embodiments of the invention, DMA process will be initiated by this register. When CPU accesses this register, DMA request will be queued to DMA request FIFO. Channel bits indicates which DRAM transaction needs to be done.

TABLE A14: DMA command register

Bit field	Name	R/W	default	Description
31:28	Channel	R/W	0	
27:23	Transfer size [4:0]	R/W	0	
22:0	DRAM address [22:0]	R/W	0	

DMA commands are issued by the CPU during expansion operations to move blocks of data between the FIFOs on chip and the expansion buffers in external memory. In embodiments of the system, DRAM address [31:23] is pointed by bit [8:0] of data buffer base address register. Transfer size register determines how many word to transfer from/to DRAM. If Transfer size[4:0] = 5'b11111, threshold value defined by threshold register 1-3 are used for DRAM transfer size. Otherwise, this register value multiplied by 4 is used for DRAM transfer size. Minimum transfer size is 4 (Transfer size[4:0] = 5'b00001) and maximum transfer size is 120(Transfer size[4:0] = 5'b11110), step size is always 4. Channel bits indicates following transactions. When DMA command is issued, corresponding interrupt is negated. After DMA command is granted and transfer is done, interrupt logic is enabled.

For example, Read Free FIFO DMA command is issued by CPU, Read Free FIFO interrupt is disabled by hardware since this interrupt procedure is handled or performed. After Read Free FIFO DMA request is popped out of this DMA request FIFO and DMA transaction finished, Read Free FIFO interrupt is enabled.

TABLE A15: DMA command detail

Bit31	Bit30	Bit29	Bit28	Channel
0	0	0	0	Rx Ready Write FIFO expansion DMA (write)
0	0	0	1	Rx Ready Read FIFO expansion DMA (read)
0	0	1	0	Read Free FIFO DMA (read)
0	0	1	1	Tx 1 Ready Write FIFO expansion DMA (write)
0	1	0	0	Tx 1 Ready Read FIFO expansion DMA (read)
0	1	0	1	Tx 2 Ready Write FIFO expansion DMA (write)
0	1	1	0	Tx 2 Ready Read FIFO expansion DMA (read)
0	1	1	1	Tx 3 Ready Write FIFO expansion DMA (write)
1	0	0	0	Tx 3 Ready Read FIFO expansion DMA (read)
1	0	0	1	Tx 4 Ready Write FIFO expansion DMA (write)
1	0	1	0	Tx 4 Ready Read FIFO expansion DMA (read)
1	0	1	1	Write Free FIFO A DMA (write)
1	1	0	0	Write Free FIFO B DMA (write)
1	1	0	1	Write Free FIFO C DMA (write)
1	1	1	0	Write Free FIFO D DMA (write)
1	1	1	1	reserved

TABLE A16: BRQ expansion counter register

Bit field	Name	R/W	default	Description
15:0	BRQ expansion counters	R/W	0	DRAM transaction counter value For monitoring purpose

TABLE A17: TRQ expansion counter register

Bit field	Name	R/W	default	Description
15:0	TRQ expansion counters	R/W	0	DRAM transaction counter value For monitoring purpose

TABLE A18: Current DMA register

Bit field	Name	R/W	default	Description

Bit field	Name	R/W	default	Description
31:28	Channel	R	0	
27:0	DRAM address [27:0]	R	0	

This register selects FIFO to monitor its status. Only one bit can be set to "1".

TABLE A19: FIFO status FIFO pointer

Bit field	Name	R/W	default	Description
14:0	FIFO pointer	R/W	0	Bit 0 = 1 : RFQ FIFO Bit 1 = 1 : BRQ write FIFO Bit 2 = 1 : BRQ read FIFO Bit 3 = 1 : TRQ write FIFO 1 Bit 4 = 1 : TRQ write FIFO 2 Bit 5 = 1 : TRQ write FIFO 3 Bit 6 = 1 : TRQ write FIFO 4 Bit 7 = 1 : TRQ read FIFO 1 Bit 8 = 1 : TRQ read FIFO 2 Bit 9 = 1 : TRQ read FIFO 3 Bit 10 = 1 : TRQ read FIFO 4 Bit 11 = 1 : WFQ FIFO 1 Bit 12 = 1 : WFQ FIFO 2 Bit 13 = 1 : WFQ FIFO 3 Bit 14 = 1 : WFQ FIFO 4

TABLE A20: FIFO status data

Bit field	Name	R/W	default	Description
25:18	FIFO read address	R	0	
17:10	FIFO write address	R	0	
9:2	FIFO level	R	0	
1	Full	R	0	
0	Empty	R	1	

Initiate data pointer read operation for TRQ (data pointer) read through CBUS. Setting bit 31 initiates read operation for TRQ, this bit is automatically reset. This function is not be used with normal I/O core request.

TABLE A21: TRQ pointer read

Bit field	Name	R/W	default	Description
31	Start read	R/W	0	Initiate data pointer read operation from TRQ. Reset after this operation finishes.
30:27	FIFO empty[3:0]	R	1111	"1" when TRQ write FIFO empty & TRQ read FIFO empty
26:25	Tx_DP_ID[1:0]	R/W	0	Select TRQ.
24:16	Reserved			
15:0	Tx_DP[15:0]	R	0	Data pointer. Updated after read operation finishes.

The BRQ Framer Counter is only used as a debugging tool to keep track of how many received frames have not been read out by the CPU. This counter indicates the number of frames of BD stored in RRQ FIFO. This counter is increment when Rx_frame_cmp comes from I/O core, decrement when BD with type[1:0] = 2'b00 or type[1:0] = 2'b11 read out by CPU.

TABLE A22: BRQ frame counter

Bit field	Name	R/W	default	Description
31:10	Reserved			
9:0	Rx frame counter	R	0	

FIFO illegal condition causes register reports which FIFO condition causes FIFO illegal condition interrupt. Only the very first illegal condition is reported. If multiple FIFO illegal conditions happen exactly at the same time, these multiple causes are reported. All the bits in this register are reset when FIFO illegal condition interrupt bit (bit[17]) of interrupt status register is set to "1".

TABLE A23: FIFO illegal condition cause register

Bit field	Name	R/W	default	Description
28	FIFO Illegal Interrupt cause	R	0	MSB of expansion counter goes "H"
27		R	0	TRQ 4 read FIFO full and DMA push
26		R	0	TRQ 3 read FIFO full and DMA push
25		R	0	TRQ 2 read FIFO full and DMA push
24		R	0	TRQ 1 read FIFO full and DMA push

Bit field	Name	R/W	default	Description
23		R	0	TRQ 4 write FIFO empty and DMA pop
22		R	0	TRQ 3 write FIFO empty and DMA pop
21		R	0	TRQ 2 write FIFO empty and DMA pop
20		R	0	TRQ 1 write FIFO empty and DMA pop
19		R	0	RRQ read FIFO full and DMA push
18		R	0	RRQ write FIFO empty and DMA pop
17		R	0	TRQ 4 write FIFO full and CPU push
16		R	0	TRQ 3 write FIFO full and CPU push
15		R	0	TRQ 2 write FIFO full and CPU push
14		R	0	TRQ 1 write FIFO full and CPU push
13		R	0	RRQ write FIFO full and I/O push
12		R	0	WFQ 4 empty and DMA pop
11		R	0	WFQ 3 empty and DMA pop
10		R	0	WFQ 2 empty and DMA pop
9		R	0	WFQ 1 empty and DMA pop
8		R	0	RFQ full and DMA push
7		R	0	WFQ 4 full and CPU push
6		R	0	WFQ 3 full and CPU push
5		R	0	WFQ 2 full and CPU push
4		R	0	WFQ 1 full and CPU push
3		R	0	WFQ 4 full and I/O push
2		R	0	WFQ 3 full and !/O push
1		R	0	WFQ 2 full and I/O push
0		R	0	WFQ 1 full and I/O push

I/O Interface

Table A24 includes description of I/O interface signals for buffer manager 100.

TABLE A24: I/O interface signals

Number	Signal	I/O	Description
1	Rx_DP_mty	Output	No RFQ available
2	Rx_DP_PTR[15:0]	Output	RFQ data output to I/O core.
3	Rx_PTR_taken	Input	Current RFQ taken, advance to next pointer
4	Rx_Rtn_PTR[15:0]	Input	Data pointer , goes to RRQ.

Number	Signal	I/O	Description
5	Rx_type[1:0]	Input	2 bit type information 00 : Single BD 01 : First of BD chain 10 : Middle of BD chain 11 : Last of BD chain
6	Rx_buff_cmp	Input	Current BD complete, update data pointer and type for RRQ
7	Rx_frame_cmp	Input	Current packet complete, update status, data length and source/destination address for RRQ
8	Rx_Rtn_ack	Output	Rx_buff_cmp acknowledge
9	Rx_length[10:0]	Input	Received packet length
10	Rx_Len_src_adr[4:0]	Input	Source address(MAC) 5'b00001 (USB0) 5'b00010 (USB1) 5'b00100 (Serial port0) 5'b01000 (Serial port1) 5'b10000 (Serial port2) Rx_length[15:11] (ATM)
11	Rx_VC_dst_adr[4:0]	Input	Destination address (MAC) Virtual Channel (ATM) 5'b00000 (All the other I/O core)
12	Rx_status[6:0]	Input	TBD
13	Rx_ID[1:0]	Input	Identifier information
14	Tx_DP_mty[3:0]	Output	TRQ empty
15	Tx_TRQ_ID[1:0]	Input	TRQ ID during request
16	Tx_DP_PTR[15:0]	Output	Current TRQ data pointer
17	Tx_length[15:0]	Output	Transmit packet data length
18	Tx_ID[1:0]	Output	ID output from TRQ
19	Tx_Type[1:0]	Output	2 bit type information for I/O core from TRQ
20	Tx_VC_ID[4:0]	Output	Virtual Channel for ATM core
21	Tx_DP_Req	Input	TRQ data pointer request
22	Tx_ack	Output	TRQ transmit acknowledge
23	Tx_Rtn_PTR[15:0]	Input	Data pointer, goes to WFQ
24	Tx_Rtn_ID[1:0]	Input	Return ID input from I/O core
25	Tx_buff_cmp	Input	Current BD complete, update data pointer and ID for WFQ
26	Tx_Rtn_ack	Output	Tx_buff_cmp acknowledge
27	OAM_FCS	Output	FCS output when MAC, OAM output when ATM
28	IO_reset	Output	Reset signal to I/O core, derived from control register, lstate.
29	IO_reset_done	Input	Reset complete signal from I/O core.
29	Base_adr[8:0]	Output	Data buffer base address, direct output from data buffer base address register to I/O module.
30	Tx_spare[3:0]	Output	Tx_spare output from TRQ FIFO

FIG. 14 illustrates a block diagram of buffer manager 100 and the interface signals in accordance with one embodiment of the system.

MBUS Interface

Table A26 provides a description of MBUS interface signals, in accordance with one or more embodiments of the system.

TABLE A26: MBUS interface signals

Number	Signal	I/O	Description
1	BREQi	Output	Bus Request
2	BGNTi	Input	Bus Grant
3	ADS	Output	Address Strobe
4	ADDR[31:0]	Output	Address. The first cycle contains the full byte address; the second cycle contains other pertinent information.
5	RSS	Input	Response Strobe
6	RSP[7:0]	Input	Response; driven by DMA controller. RSP[7] : Direction 0: read, 1 : write RSP[6:2]: Bus Agent ID RSP[1:0]: transfer tag
7	DATA[31:0]	Inout	32-bit data either driven by the DMA controller (memory read), or the bus agent (memory write).
8	BS[1:0]	Output	Bank select signal. DRAM address [22:21]

APPENDIX B:

BUFFER MANAGER INTERFACE WITH WAN ENGINE

The WAN engine comprises of a RISC CPU, a set of hardware assist logic, and the appropriate WAN protocol firmware. The WAN engine handles the segmentation and assembly of communication packets. The WAN engine interfaces with buffer manager 100 via a set of coprocessor registers. A coprocessor is a secondary processor used to speed up operations by handling some of the workload of the main CPU. Embodiments of the invention are described herein as applicable to a RISC processor and a WAN coprocessor based on a MIPS architecture. This description is provided by way of example, however, and should not be construed as a limitation. Other embodiments of the invention may be implemented using other processor and coprocessor architectures. The following tables list the control and status registers of a WAN coprocessor and CBUS interface 260. In accordance with one aspect of the invention, the control registers map to the set of 32-bit coprocessor control register space of the MIPS architecture, for example.

TABLE B1: Exemplary Coprocessor (CP) Control Register

CP Control Register Number (Hex)	Register	Description
0	WAN_control	WAN control register. (CBUS accessible)
1	WAN_status0	Status bits
2	WAN_status1	
3	WAN_DMA_cmd	Write port of the 8 words deep DMA command queue.
4	WAN_DMA_addr	Write port of the 8 words deep DMA address queue.
5	WAN_DMA_base_addr	DMA base address
6	WAN_rx_cmd	Write port of the 8 words deep WAN receive command FIFO
7	WAN_tx_cmd	Write port of the 8 words deep WAN transmit command FIFO
8	WAN_rdfrq	RFQ pointer
9	WAN_rxbdq0	RRQ 1
A	WAN_rxbdq1	RRQ 2
B	WAN_txbdq0	TRQ 1
C	WAN_txbdq1	TRQ 2
D	WAN_wrfrq	WFQ

CP Control Register Number (Hex)	Register	Description
E	WAN_rx_cmd_done	Receive command Done Queue (8 deep)
F	WAN_tx_cmd_done	Transmit command Done Queue (8 deep)
0x10	WAN_DMA_cmd_done	DMA command Done Queue (8 deep)
0x11	WAN_CRC_cmd	
0x12	WAN_CRC_input	
0x13	WAN_CRC_state0	
0x14	WAN_CRC_state1	
0x15	WAN_IDLE_TX_CMD	Idle cell transmit command
0x16	WAN_IDLE_CNT	Idle cell transmitted counter
0x17	WAN_interrupt_status	Written by WAN-RISC to generate interrupt to software RISC (CBUS accessible)
0x18	WAN_mask	Mask register (CBUS accessible)
0x19	WAN_cmd	Command register (CBUS accessible)
0x1A	WAN_Data	Data register (Cbus accessible)
0x1B	WAN_DMA_address	Dma address register. (Cbus accessible)
0x1C	WAN_frame_sizes	Frame and delimiter sizes
0x1D	WAN_frame_delimiter	Frame delimiter
0x1E	WAN_frame_delimiter_mask	Frame delimiter mask
0x1F	WAN_frame	WAN frame info

TABLE B2: Exemplary CBUS Register

CBUS Register Number (Hex)	Register	Description
0x0 0x4	WAN_control WAN_interrupt_status	WAN control register. (CP accessible) Written by WAN-RISC to generate interrupt to software RISC (CP accessible)
0x8	WAN_mask	Mask register (CP accessible)
0xc	WAN_cmd	Command register (CP accessible)
0x10	WAN_Data	Data register (CP accessible)
0x14	WAN_DMA_address	Dma address register. (CP accessible)
0x18	VCXO_period0	See NV_vcxo_controller spec. CBUS access only
0x1c	VCXO_count0	CBUS access only
0x20	VCXO_period1	CBUS access only
0x24	VCXO_count1	CBUS access only

TABLE B3: Exemplary WAN Control Register

Bit field	Name	R/W	default	Description

Bit field	Name	R/W	default	Description
31:30	lstate	R/W	2'b10	00:run; 11;halt; (default) 10:reset, reset the WAN Interface
29	Rx_reset	R/W	0	Reset receive interface – this is a 1-shot that is set by software and reset by hardware
28	Tx_reset	R/W	0	Reset transmit interface – this is a 1-shot that is set by software and reset by hardware
27	Debug_Write_enable	R/W	0	Write enable for bits 26:24
26:24	Debug	R/W	0	For debug purposes only
23:11	Reserved	R	0	
10:9	WAN_configure	R		Read only 00 : UTOPIA 01 : Serial, non-HDLC 11 : Serial, HDLC 10: Terayon
8	Write_enable for bit 7	R/W		Read back 0
7	Busy_bit	R/W	1'b0	When SW MIPS writes to WAN_cmd register, this bit will be set by hardware. WAN MIPS will clear it when the command is done
6	Write_enable for[5:3]	R/W		Read back 0.
5	ODD_PRTY	R/W	1'b0	1 : odd parity, 0 : even parity
4	EN_UTOPIA_PRTY	R/W	1'b0	Enable Parity for UTOPIA Interface
3	EN_IDLE_INSERTION	R/W	0	This bit controls the behavior of the interface hardware if the transmit queue is empty. If set hardware will insert idle cells.
2	EN_TX_TRQ_ID	R/W	0	Enable bit to write transmit ready queue. If one it will update bits 1:0 else do nothing.
1:0	TX_TRQ_ID	R/W	00	ID for reading Transmit ready queue.

On power-up, the *lstate* is in "halt" state (e.g., "11"). Software running on the MIPS programs the *lstate*'s 2 bits to "run" state (e.g., "00") to start the WAN core. The WAN CORE is the on-chip hardware logic that performs the function of integrating the FWAN CPU with the rest of the FWAN hardware assist modules as well as to he system outside of the FWAN module. The MIPS software can program these bits to 2'b10 to reset the WAN core and the coprocessor interface. On completion of the reset sequence, the hardware will set *lstate* to back to "halt" (default) state. When a reset is issued, all pending WAN and DMA commands in

the FIFO 230 queues will be flushed. A DMA command that has already started (e.g., if data transfer is in progress or if a memory request is already issued by the DMA command sequencer) will be allowed to complete. Thus, the reset sequence will wait until the transfer is finished.

Table B4 describes the contents of the FWAN Control Register. This register is primarily used to determine the status of specific command and command done queues as well as to determine when a queue command request has been acknowledged. The acknowledge status bits are used to indicate to the FWAN CPU that a read or write to a command queue is permitted. This allows previous commands to complete execution before new commands are entered.

TABLE B4: WAN Status 1 Register

Bitfield	Name	R/W	default	Description
31:16	reserved			
14	Dma_cmd_doneq_empty	R	1	DMA command done queue empty
13	Dma_cmdq_full	R	0	DMA command queue full
12	Tcmd_doneq_empty	R	1	Transmit command done queue is empty.
11	Tcmdq_full	R	0	DMA done command queue is full.
10	Rcmd_doneq_empty	R	1	Receive done command queue is empty.
9	Rcmdq_full	R	0	Receive command queue is full
8	TFQ_ACK	R	1	Transmit free queue acknowledge.
7	TX_ACK	R	1	TRQ ack. 1 = can read from TXRD[1:0] 0 = cannot read from TXRD[1:0] - BM pre-reading the queue.
6:3	TRQ_empty	R	0xf	TRQ status 1: Empty 0: Non-empty bit 3:TRQ0, bit 4:TRQ1, bit 5:TRQ2, bit6:TRQ3
2	Reserved	R	0	
1	RX_RTN_ACK	R	1	RRQ acknowledge 1 = can write into RXRD[1:0] register 0 = cannot write into RXRD[1:0] register (previous write in progress).
0	Rfrq_empty	R	1	RFQ fullness 1=empty, 0=non-empty

Table B5 contains the descriptions of the command queue fullness registers. These registers are used to inform the FWAN CPU whether a particular command queue is full or empty. If a command queue is deemed not empty, then the CPU is free to queue a new command. Similarly, if a command done queue is not empty, the CPU should proceed to read out the contents of the queue until it is empty.

TABLE B5: WAN Status 0 Register

Bit field	Name	R/W	default	Description
25:22	Tcmd_done_qfullness	R	0	Transmit command done queue fullness 0 = empty; 4 = full
21:18	Rcmd_done_qfullness	R	0	Receive command done queue fullness 0 = empty; 4 = full
17:13	Dma_done_qfullness	R	0	DMA command done queue fullness 0 = empty; 8 = full
12:8	Dma_qfullness	R	0	DMA command queue fullness 0=empty; 8=full
7:4	transmit_qfullness	R	0	Transmit command queue fullness 0=empty; 4=full
3:0	receive_qfullness	R	0	Receive command queue fullness 0=empty; 4=full

With respect to RFQ 232, the status register includes an empty flag to indicate whether RFQ 232 is non-empty. A read to the co-processor register RFQ_DP returns a 16-bit free data pointer. If the queue is empty the control software does not read RFQ 232 to receive a data pointer.

With respect to RRQ 234, two coprocessor registers (WAN_rxbdq[1:0]) are utilized to provide 46-bits needed for storing a 46-bit concatenated buffer descriptor, for example. Writing to WAN_rxbdq1 causes the contents of WAN_rxbdq[1:0] to be written to the RRQ 234. Writing to WAN_rxbdq0 updates the register. The writing takes place after the buffer manager acknowledges the previous write. Control software examines the RX_RTN_ACK bit in the WAN_status0 register before writing WAN_rxbdq[1:0] register. In the worst case scenario, an ACK is returned in 12 cycles from a previous write to WAN_rxbdq[1:0] register. If writes to WAN_rxbdq[1:0] register are more then 12 cycles apart then there is no need to

examine the RX_RTN_ACK bit. Control software also examines RRQ 234 to ensure that RRQ 234 is not full before writing to WAN_rxbdq[1:0] register.

The 64-bits included in WAN_rxbdq[1:0] are concatenated to 46-bits so that they can be stored in RRQ 234. Because the bits in RRQ 234 are interpreted by control software, any content can be forwarded from the WAN RISC to the main CPU. If more than 46-bits are required, then multiple buffer descriptors can be send to convey the information.

With respect to TRQ 236, where four transmit queues are included, when any of the four transmit queues is non-empty the empty flag will be de-asserted. This will cause an interrupt to the WAN RISC processor to indicate there is at least one entry in one of TRQ 236s. The four empty bits are provided in the WAN_status register. The four transmit queues hold different priority buffer descriptors. The priority algorithm is implemented in software executed by the WAN RISC, in accordance with one or more embodiments of the invention.

The WAN RISC sets the Tx_TRQ_ID (identifying which transmit buffer to read) field in the WAN_control register which starts a read from one of the transmit queues. Once the read is performed, then buffer manager 100 asserts a TX_ACK and writes the transmit queue's content to two co-processor registers WAN_txbdq[1:0]. After the ACK is asserted, WAN RISC reads the WAN_txbdq0 and WAN_txbdq1 co-processor register to get the contents of TRQ 236. Control software examines the TX_ACK bit in the WAN_status register before reading the WAN_txbdq[1:0] registers to verify that the BD in the txbdq register is valid. In accordance with an aspect of the invention, the maximum wait to receive an ACK is 12 cycles from the time the ID register is set because, in embodiments of the invention it takes at least 12 cycles to select the TRQ 236 to read from and to retrieve the BD. Thus, there is no need to examine the ACK bit if the read is be delayed by 12 cycles. Similar to WAN_rxbdq[1:0], WAN_txbdq[1:0] has 46-bits, in certain embodiments of the system. Any information can be passed from the main RISC to the WAN RISC as the bits are interpreted by control software.

With respect to WRQ 238, control software frees a data pointer by writing into the TX_RTN_PTR co-processor register. In embodiments of the system, buffer manager 100's hardware determines the queue in which the pointer will be stored. Thus, thee firmware does not control this operation. If the WRQ 238 is full, the hardware will not issue an ACK. By not issuing an ACK the software cannot free the Data Pointer because that indicates that the WFQ 238 is full. The software does not free the data pointer unless an ACK is received from a previous write.

Table B6 describes the registers used by the FWAN CPU to obtain a free data pointer and to write a BD into RRQ 234. Because each BD is comprised of 8 bytes, their two registers are used in accordance with one or more aspects of the system. One register handles the upper 4 bytes and the other register handles the lower 4 bytes of the BD.

TABLE B6: WAN_rdfrq

Bit field	Name	R/W	Default	Description
15:0	RFQ_DP	R	X	Ready Free Queue pointer

TABLE B7: WAN_rxbdq0

Bit field	Name	R/W	Default	Description
31:0	RXRD0	R/W	X	Receive Ready Queue 0

TABLE B8: WAN_rxbdq1

Bit field	Name	R/W	Default	Description
31:0	RXRD1	R/W	X	Receive Ready Queue 1

The following shows the format of rxrd0 and rxrd1. Reserved bits are thrown away by hardware when writing to Buffer Manager. Software can interpret other bits any way it wants. RxRd0[19:18] is hardwired to 2'b11. Buffer Manager FIFO does not contain these 2 bits.

Table B9 : Rxbdq0/1 mapping

Word	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Rxbdq0[31:16]	Virtual Channel					OAM	status						2'b11		Type	
Rxbdq0[15:0]	Data Length (total in all data buffers)															
Rxbdq1[31:16]	Data buffer base address offset									Data Pointer [15:9]						
Rxbdq1[15:0]	Data Pointer [8:0]									7b'0000000						

TABLE B10: WAN_txbdq0

Bit field	Name	R/W	Default	Description
31:0	TXRD0	R	X	Transmit Ready Queue 0

TABLE B11: WAN_txbdq1

Bit field	Name	R/W	Default	Description
31:0	TXRD1	R	X	Transmit Ready Queue 1

The following shows the format of txrd0 and txrd1. Some bits are hardwired to 1 or 0. Software can interpret other bits any way it wants. Note that Txbdq0[19:18] is contained in Buffer Manager FIFO. Txbdq0[21:20] is not contained in buffer manager FIFO. For the Final BD, software can interpret the bit fields any way it wants. The reserved bits, data buffer address offset and Txbdq1[6:0] will not be inside the buffer manager FIFO.

TABLE B12: Txbdq0/1 mapping

Word	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Txbdq0 [31:16]	Virtual Channel					OAM	status				Reserved		Identifier		Type	
Txbdq0 [15:0]	Data Length (total in all data buffers)															

Txbdq1 [31:16]	Data buffer base address offset	Data Pointer [15:9]
Txbdq1 [15:0]	Data Pointer [8:0]	7'b0000000

TABLE B13: WAN_wrfrq

Bit field	Name	R/W	Default	Description
17:16	TX_FREE_ID	R/W	X	Transmit Free queue ID
15:0	TX_RTN_PTR	R/W	X	Transmit Free queue pointer